

Design and implementation of a countermovement jump performance estimation system using a wearable device with IMUs based on machine learning algorithms

Jhe-Sheng Yang¹ and
Jun-Zhe Wang^{2,*}

¹Department of Computer Science,
National Yang Ming Chiao Tung
University, Hsinchu 300093, Taiwan

²Department of Computer Science
and Information Engineering,
Graduate School of Engineering
Science and Technology, National
Yunlin University of Science and
Technology, Yunlin 640301, Taiwan

*E-mail: jzwang@yuntech.edu.tw

Received for publication
April 04, 2025

Abstract

The rapid evolution of technology has led to a proliferation of wearable devices. These devices have a wide range of applications in hospital health care, health monitoring, and sports. In sports science, it is possible to assess an athlete's neuromuscular function via explosive power in the legs, and the most common test for this is the counter-movement jump (CMJ). Traditionally, to assess the explosive power of an athlete's legs, researchers ask the athlete to stand on a force plate and perform a CMJ several times. From this, seven variables are calculated using the force-time curve measured by the force plate, and then these variables are interpreted by sports professionals to evaluate the goodness of the CMJ. However, a force plate is expensive and difficult to transport. Taking this into account, this paper designs a machine learning-based model to predict CMJ performance variables via an economical and wearable device with inertial measurement units (IMUs). The experimental results demonstrate that the proposed method is capable of predicting most performance variables of CMJ with an acceptable rate of error. Finally, we also implement a real-time performance variable measurement system to demonstrate the applicability of the proposed method.

Keywords

wearable device, machine learning, inertial measurement unit, force plate, countermovement jump

1. Introduction

Wearable devices with inertial measurement units (IMUs) are increasingly used in health care [6, 17, 23], health monitoring [12, 32, 33], and sports [4, 13, 14]. Companies, such as Garmin¹, Fitbit², and Suunto³, sell affordable wearable devices for general sporting activities; however, athletes involved in competitive

sports exist at the cutting edge of health care, physiological optimization, and technology. One important sport-science metric used to assess the explosive power in an athlete's legs is the countermovement jump (CMJ). Measurement of CMJ is important because it involves the detection of key physiological variables such as flight time, peak force, and average loading rate. Because this particular mixture of

¹ <https://www.garmin.com/en-US/c/sports-fitness/activity-fitness-trackers/>

² <https://www.fitbit.com/global/us/products>

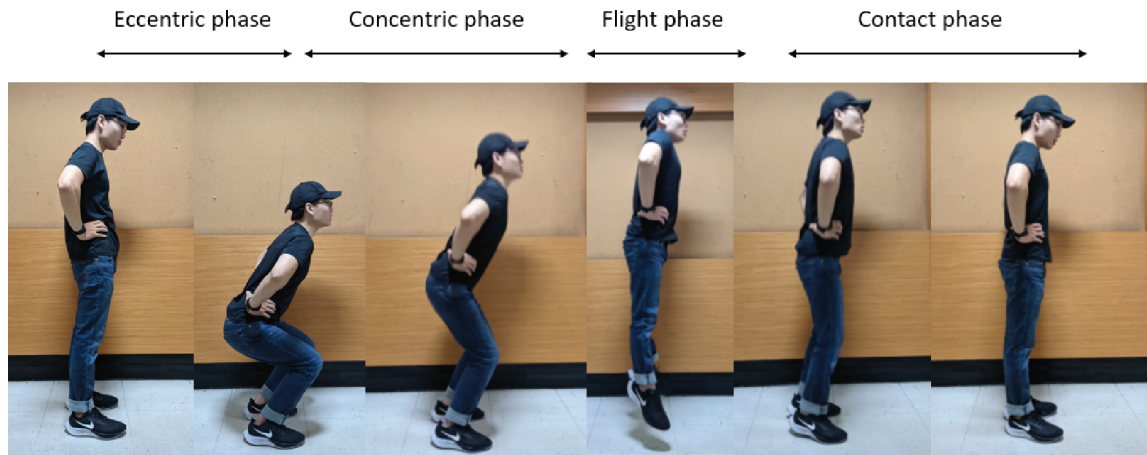
³ <https://www.suunto.com/Product-search/See-all-Sports-Watches/>

variables encompasses so many athletic and physiological properties, CMJ (or the variables) can be used to assess the general physical condition and neuromuscular function of an athlete.

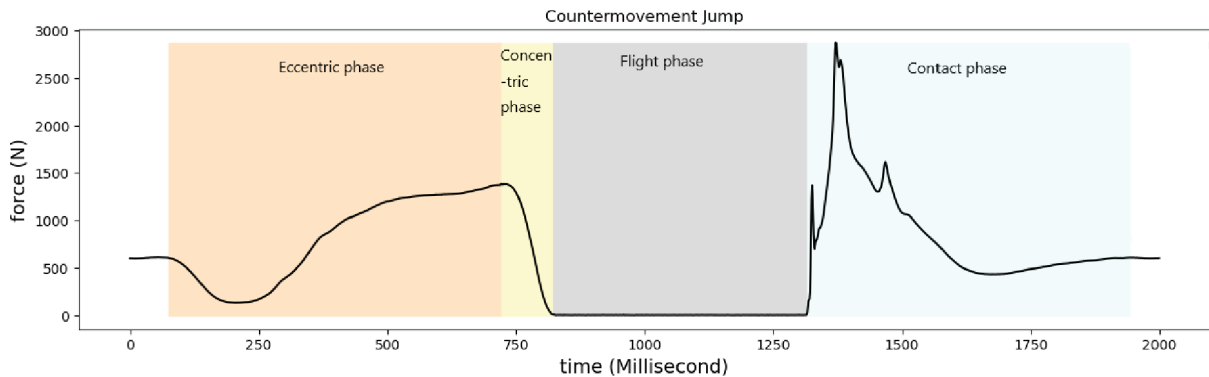
Conventionally, logging the variables required to compute a CMJ value requires an athlete to perform a CMJ several times while standing on a force plate. The corresponding force-time data are measured by the force plate, and the variables are computed from those observations. Sport professionals can evaluate the quality of a CMJ by utilizing these variables. The primary shortcoming of this system is the size, weight, and expense of the equipment involved. Laboratory quality force plate systems range from £30,000 GBP to £70,000 GBP, and (somewhat) portable equivalents range from £10,000 GBP to £15,000 GBP [20]. Conversely, IMUs for consumer or industrial use cost from US\$10 to US\$1,000 [2]. Compared to professional tier systems, a wearable device that

can detect similar variables utilizing IMUs would have the benefit of being considerably cheaper, lighter, and more mobile. Given the aforementioned aspects, the motivation of this study is to design a wearable, yet economical, CMJ performance variable prediction model, using off-the-shelf IMUs. The performance variables of interest to assess the CMJ include *peak force*, *second peak force*, *flight time*, *average loading rate*, *net impulse*, *peak power*, and *rate of force development (RFD)*. To show that this system can be deployed in a real-world usable situation, this study also creates a prototype using a wearable IMU that can automatically predict the seven CMJ performance variables, after a test subject has performed a series of CMJs.

As shown in Figure 1A, there are four key CMJ phases: *eccentric*, *concentric*, *flight*, and *contact*. In the *eccentric* phase, the jumper is standing on a force plate with their hands on their waist, squatting



(a) The entire counter-movement jump process.



(b) Force-time data corresponding to the CMJ shown in Figure 1a.

Figure 1: The process and corresponding force-time data of a CMJ. (A) The entire CMJ process. (B) Force-time data corresponding to the CMJ shown in (A). CMJ, counter-movement jump.

until their knees are maximally bent. The *concentric* phase begins when the body has bounced, passing its lowest point, but has not yet left the ground. Once the feet leave contact with the ground, the *flight* phase begins, and the process concludes with the *contact* phase as the jumper lands. Conventionally, the related force-time data are recorded by the force plate, as shown in Figure 1B. With that force-time data, the seven aforementioned performance variables related to the jumper's physical condition can be computed, as shown in Figure 2.

To accomplish our goal, the major challenge we need to face is how to use the IMU sensing data (e.g., acceleration-time data) to accurately predict the CMJ performance variables calculated from the force-time data measured by a force plate. To address this problem, we tie a wearable device with IMUs around a jumper's waist and ask her or him to stand on a force plate to perform a CMJ several times. After a series of CMJs is performed, we collect the acceleration-time and force-time records measured by the wearable device and the force plate, respectively. Then, the seven performance variables can be derived from the force-time record of the force plate as the ground truth values. We then extract some jump performance-related features from the acceleration-time

data measured by the IMUs for training a model to fit the performance variables. Since the seven performance variables we want to estimate are numerical, we model the problem of predicting a different performance variable as an independent regression problem, only relying on the related features extracted from the acceleration-time data of wearable IMUs. That is, we train a regression model to predict each performance variable of a CMJ obtained from force-time data as ground truth, and the input attributes (independent variables) of the regression model are the features extracted from acceleration-time data of IMUs. As a result, five regression models, such as linear regression (LR), linear support vector regression (LSVR), decision tree regression (DTR), light gradient boosting machine regression (LGBMR), and eXtreme gradient boosting regression (XGBR), are adopted and compared in our experiments. We collect 870 CMJs from 19 jumpers for experiments and compare the goodness of these regression models in terms of prediction errors. Experimental results demonstrate that LGBMR outperforms other models in predicting most performance variables.

Furthermore, to demonstrate the practicality of our method, we develop a demonstration system that receives the acceleration-time data from a

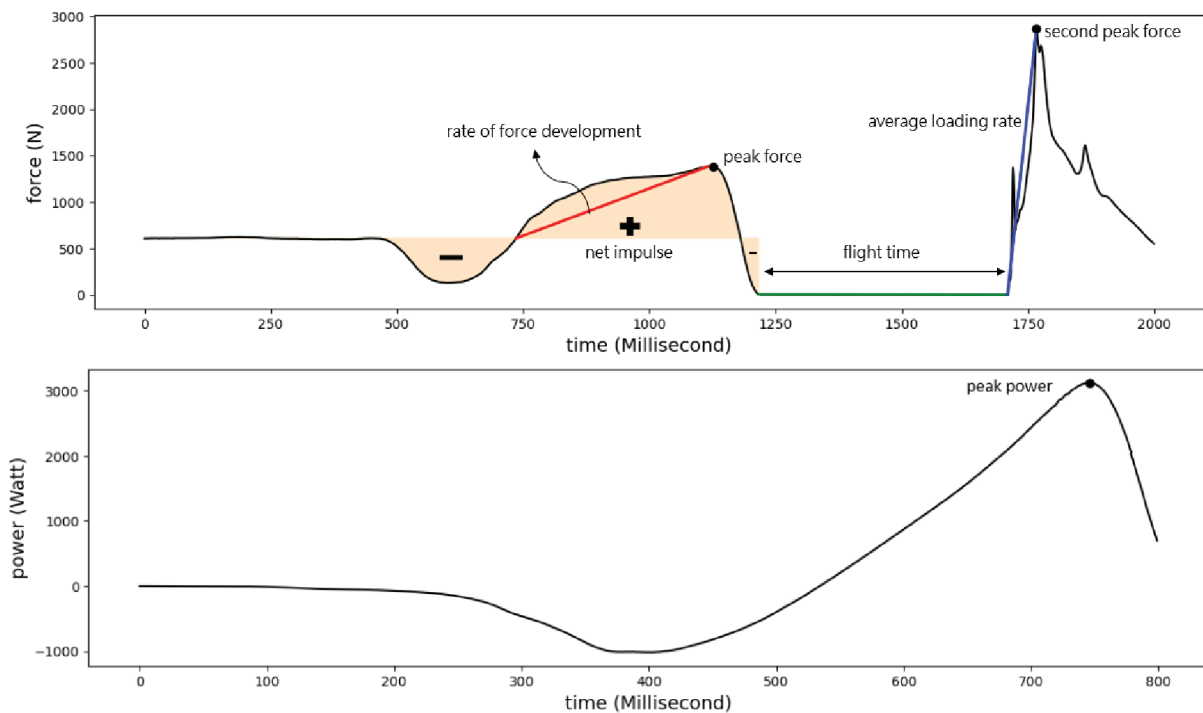


Figure 2: Seven performance variables used to assess CMJ, derived from the force-time curve and the power-time curve. CMJ, countermovement jump.

wearable device with IMUs tied to a jumper's waist via Bluetooth in real-time. After a series of CMJs is completed by the jumper, the system will immediately estimate the value of each of the seven performance variables, only relying on the acceleration-time data based on the underlying machine learning model (i.e., LGBMR in this paper). With this system, a jumper can easily obtain his/her jump performance variables through wearable IMUs instead of using an expensive and hard-to-carry force plate.

The rest of this paper is organized as follows. Section II reviews the related work. Section III describes the proposed method, while Section IV shows the experimental results. The demonstration system is given in Section V. Finally, Section VI concludes this paper.

II. Literature

a. Vertical jump in sports science

The vertical jump is a specific action often utilized in sports science to assess the general physical condition of an athlete. CMJ is one of the most common types of exercise used for general physical assessment. To measure the physical condition of a subject, a researcher first asks the subject to perform a vertical jump (e.g., CMJ) while standing on a force plate. From this, a stream of force-time data is captured and converted into a series of variables (usually 7), which reflect the subject's jumping performance and the overall physical condition. Swinton et al. [31] described how to apply the force-time curve of a CMJ and squat jump (SJ) to resistance training. They established a connection between resistance training activity, SJ, and CMJ, demonstrating a way of improving training efficiency. Rice et al. [29] examined the power-time and force-time curves for male and female basketball players to determine whether there was a significant difference between the genders. Pupo et al. [28] measured the CMJ and SJ performance of volleyball players and sprint runners, and then discussed the relation between the jump performance variables. They showed that the maximum force and the peak velocity were the main determinants of jump height and power, and that the sprint runners' jump performance was superior to volleyball players. Kirby et al. [19] discussed the relationship between jump performance variables, including net impulse, peak force, peak power, peak velocity, and jump height. The result showed that relative net vertical impulse has a significant correlation

to jump height, and peak force is negatively correlated with jump height. Lake et al. [21] used push press and loaded CMJ to quantify load and its effect on resistance training, by evaluating peak and mean power. McMahon et al. [26] researched senior and academy rugby league players as they performed CMJs. They found that senior players had superior jump height and impulse performance. McLellan et al. [25] discussed how RFD affects vertical jump performance. They studied the relationship between RFD and other vertical jump variables such as peak force and peak power in the eccentric phase. The result showed that RFD has a significant association with some of those variables.

b. Studies applying machine learning algorithms to vertical jump

Some studies have already applied machine learning to vertical jump data. Zhou et al. [34] used machine learning methods to predict football players' CMJ jump height. Their models utilized LR, random forest, and decision-tree regression. The result showed that the Pearson product-moment correlation coefficient of all their models was >0.85 . Kipp et al. [18] used a machine learning approach to predict the CMJ performance of track and field athletes after resistance training. Chiang et al. [11] proposed using wearable IMU devices instead of a force plate, to evaluate CMJ performance. They collected CMJ data from the IMU device and the force plate, then converted the acceleration data into features, and finally used the force plate data as the ground truth to train the machine learning models so that they could predict the CMJ performance. The models they utilized were decision-tree regression, LSVR, and LR. The experiment results showed that the machine learning models made good predictions for CMJ peak force, net impulse, and flight time.

c. Studies combining sensing data with machine learning algorithms

Recently, there have been many research projects relating to the cooperation of sensors and machine learning. These research papers cover a wide variety of fields, and most researchers hope to replace expensive instruments or complex measurement systems with simple, low-cost sensors and machine learning. Bauman et al. [5] did a preliminary investigation of predicting time-to-next heel strike with acceleration data and machine learning.

They used two accelerometers, and used their x- and y-axis data as the raw data. The acceleration data was first processed through a second-order 10 Hz cutoff Butterworth filter. They used the sliding window to process the acceleration data to obtain its primary features. These features were subsequently fed into artificial neural networks (ANN) and a teacher forcing ANN for training, and a grid search could then be used to find optimal model hyperparameters. Chaaban et al. [7] proposed a predictive method based on one or multiple IMU sensors to predict vertical ground force and knee biomechanics. They tied various numbers of sensors to a test subject's leg to collect the gyro and acceleration data measured by the IMUs as their model training data. In their experiments, they compared the performance difference between single and various multi-sensor setups. They determined that their setup was a valid way to replace expensive and complex laboratory force plate measurement systems.

Johnson et al. [15] proposed a method of collecting acceleration data via five sensors tied to different physical locations on a test subject's body, then using deep learning to assess the predictive quality of these data concerning multidimensional ground reaction forces while running, and sidestepping. Lim et al. [24] proposed the sacrum as the key measurement point for single IMU setups. Being near the center of mass (CoM), the sacrum can be used to effectively capture and predict dynamic data (e.g., torque of the knee, ground reaction force of the knee) of the lower limbs, by an ANN. They also showed that the CoM is a dynamic determinant of walking. Through biomechanical analysis, it is possible to approximate the subject's ground reaction force, and even the segment-angle of their lower limbs, by the weighted sum of CoM dynamics. Acceleration data were converted to velocity and displacement, and this was used to create features that can be used to predict lower limb kinetics and kinematics.

III. Methodology

This paper follows a similar system architecture to the one described in references [10, 11]. This architecture predicted each of the seven performance variables of a CMJ and operates in three phases, as shown in Figure 3.

- Phase 1 – Acceleration-data preprocessing: Segment the acceleration-time data from the IMU

into multiple individual jumps, and then extract important features from each jump.

- Phase 2 – Force data preprocessing: Normalize the force-time data from the force plate, and then segment the force-time data into multiple jumps, and calculate each value of the seven performance variables, for each CMJ.
- Phase 3 – Machine-learning model training: Select several established and recognizable features from the data features observed in Phase 1. For each of the seven performance variables, train an individual regression model to fit the value of the corresponding performance variable obtained from the force plate data.

a. Raw IMU acceleration-data preprocessing

The test subject performs several CMJs, as shown in Figure 1A. The raw acceleration-time data from the wearable IMU are logged, and Figure 4A is a plot of that data. Figure 4B shows the same data after a sixth order 30 Hz Butterworth filter has been applied to reduce noise. It is also necessary to subtract gravity (1 g) from the acceleration-time data to bring the y-axis steady-state to 0 g (Figure 4C). Finally, the normalized acceleration-time data can be divided into individual jumps (Figure 4D).

a.i Feature extraction from the processed acceleration data

Feature extraction from the preprocessed acceleration-time data for each CMJ is possible following the method proposed in references [10, 11]. This study extracts statistical features, time intervals, and critical point values, and then computes averages. The critical values from the raw acceleration-time data look like those presented in Figure 5, where point A represents the peak acceleration value, before the jumper leaves the ground, points B and D are where the acceleration value in the original acceleration-time series is zero, before normalization (i.e., 1 g in the normalized acceleration-time data), and point C is the lowest acceleration value before landing. Point E is the maximum acceleration value during the contact phase, and Point F is the second-highest acceleration value, also occurring in the contact phase, but after Point E. All of the 31 extracted features and their descriptions are listed in Table 1.

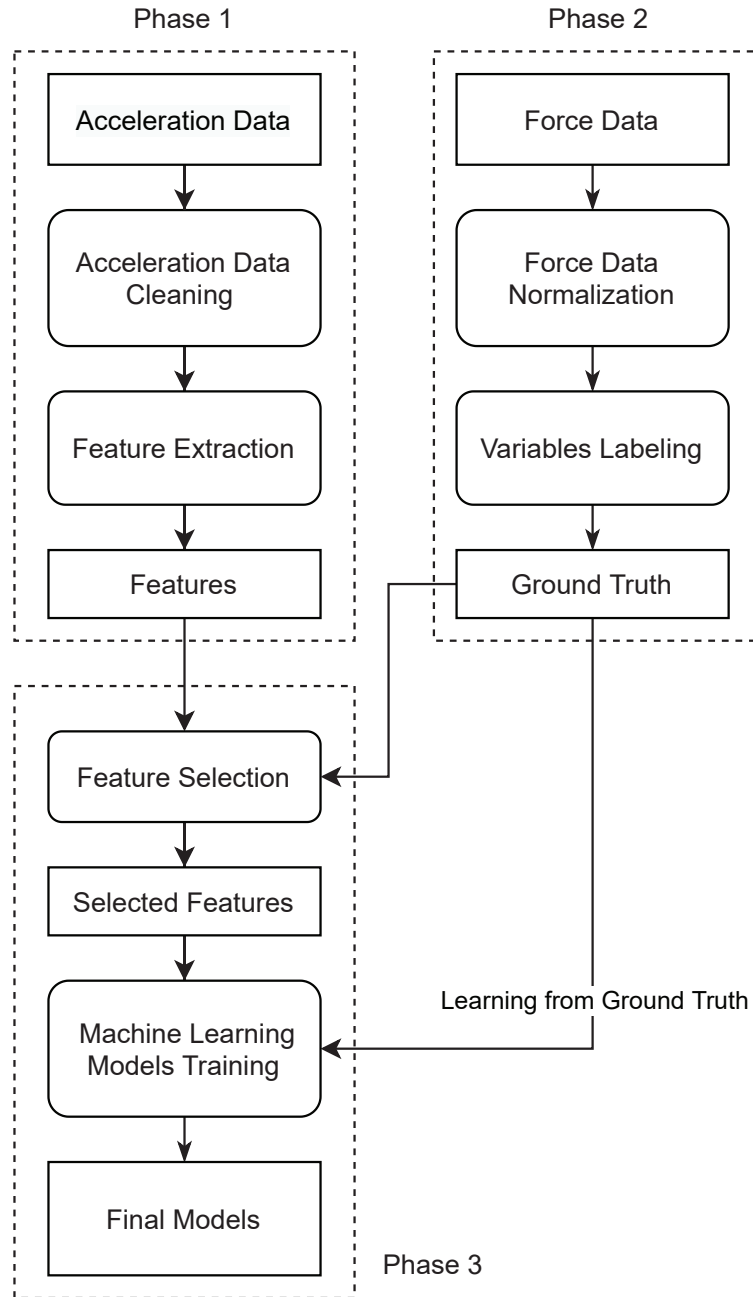


Figure 3: Proposed architecture and application workflow for a machine learning model that can predict CMJ performance variables from IMU sensor data. CMJ, countermovement jump; IMU, inertial measurement unit.

b Raw force-data preprocessing

b.i Subject weight consideration

A ground truth value is required for all CMJ performance variables to train the regression models. This ground truth is acquired by getting the test subject to perform CMJs while simultaneously wearing an

IMU-enabled device and standing on a calibrated force plate. It is necessary to acquire the IMU acceleration-time data and the corresponding force-time data from the force plate simultaneously. After a subject performs one or more CMJs, the force-time data can be plotted as shown in Figure 6A. It is intuitive that the heavier a subject is, the greater the force of

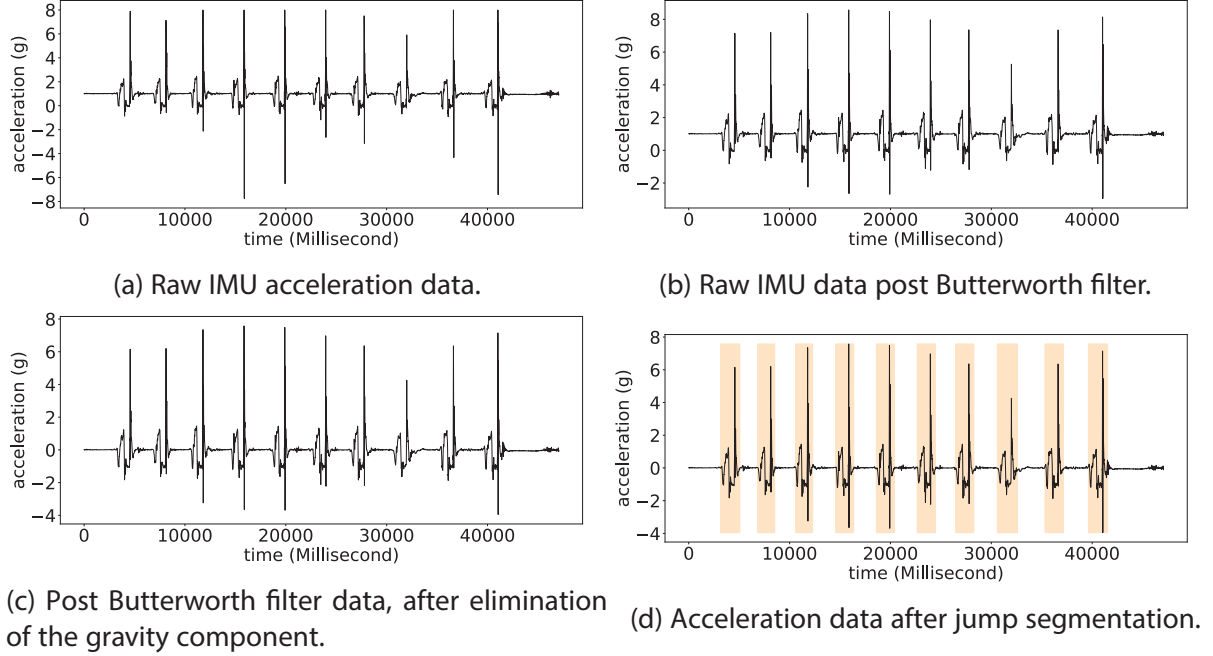


Figure 4: Data preprocessing of CMJ acceleration data. (A) Raw IMU acceleration data (B) Raw IMU data post-Butterworth filter. (C) Post-Butterworth filter data, after elimination of the gravity component. (D) Acceleration data after jump segmentation. CMJ, countermovement jump; IMU, inertial measurement unit.

their CMJ. To make the machine learning model reliable for a variety of users with different body masses, the force-time data have to be normalized to eliminate the body mass effects. Normalization is completed by dividing the raw force data by the subject's weight, as shown in Figure 6B.

c. Raw power data preprocessing

Next, it is necessary to compute the seven performance variables to derive ground truth values for training the machine learning models. Six of the performance variables are derived from the normalized force-time data, as shown in Figure 2. These variables include *net impulse*, *rate of force development (RFD)*, *peak force*, *flight time*, *average loading rate*, and *second peak force*; only the *peak power* variable is derived from the *power-time* data.

The lack of power-time data means that it is necessary to derive the power-time data from the raw force-time data by the use of Eqs (1)–(3); where t represents time, $a(t)$ represents acceleration at time t , $F(t)$ represents force at time t , m is mass, $v(t)$ is the velocity at time t , and $P(t)$ represents power at time t , measured in Watts. Since the subject weight affects the power value, the machine learning system has

to take the maximal *peak power* value, divided by the subject weight to create a Watt/kg value. This is used to eliminate the effect of subject weight on *peak power*, and Figure 7 shows the power-time data before and after division by the subject weight.

$$a(t) = \frac{F(t)}{m} \quad (1)$$

$$v(t) = \int a(t) dt \quad (2)$$

$$P(t) = F(t) \cdot v(t) \quad (3)$$

d. Machine learning model phase

d.i Feature selection

Before utilizing the total extracted features for training, important features are selected from the 31 variables detectable from the acceleration data. Feature selection can simplify a regression model, improve model performance, and reduce the risk of overfitting. The method of feature selection is filter-based ranking [8] because of its simplicity and proven performance record across a number of applications. The filter-based ranking approach is *SelectkBest*, and this is available in the *scikit-learn*

Table 1: 31 features extracted from the acceleration data

Feature	Description
Mean	The mean of the acceleration data
SD	The standard deviation of the acceleration data
IQR	The interquartile range of the acceleration data
Skewness	The skewness of the acceleration data
Kurtosis	The kurtosis of the acceleration data
Frequency	The dominant frequency of the acceleration data
Entropy	The sample entropy of the acceleration data
value_A	The value of point A
value_C	The value of point C
value_E	The value of point E
value_F	The value of point F
duration_AC	The time interval from point A to point C
duration_BD	The time interval from point B to point D
duration_CE	The time interval from point C to point E
duration_EF	The time interval from point E to point F
slope_AL	The slope from (point A – 20 ms) to point A
slope_AR	The slope from point A to (point A + 20 ms)
slope_CL	The slope from (point C – 20 ms) to point C
slope_CR	The slope from point C to (point C + 20 ms)
slope_EL	The slope from (point E – 10 ms) to point E.
slope_ER	The slope from point E to (point E + 10 ms).
slope_FL	The slope from (point F – 10 ms) to point F
slope_FR	The slope from point F to (point F + 10 ms)
slope_AC	The slope from point A to point C
slope_CE	The slope from point C to point E
slope_DE	The slope from point D to point E
slope_EF	The slope from point E to point F.
Integration	The integration value before point C
average_A	The average value between (point A – 80 ms) and (point A + 20 ms)
average_C	The average value between (point C – 20 ms) and (point C + 180 ms).
average_E	The average value between (point E – 20 ms) and (point E + 180 ms)

IQR, Interquartile range.

software package [1]. Each CMJ performance variable is assessed to determine the significance of the relationship between the feature and the performance variable. The score function of *SelectkBest* is

specified as $f_regression$ in advance, and the significance of the relationship between a feature and its related performance variable is given by the F -score. Once the F -score of each of the 31 features is

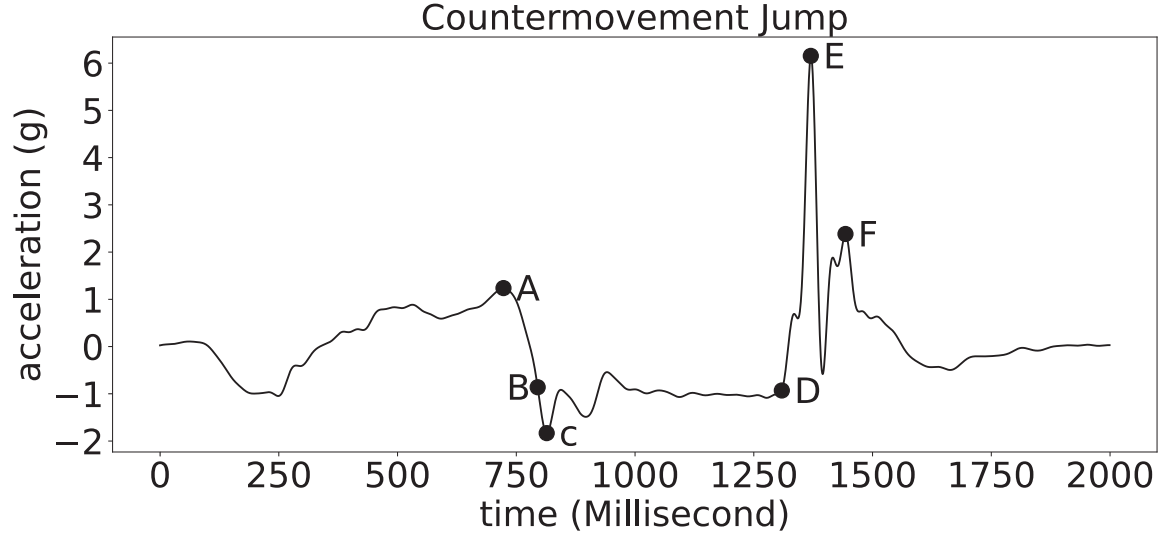


Figure 5: Critical point extraction from post-processed IMU acceleration-time data. IMU, inertial measurement unit.

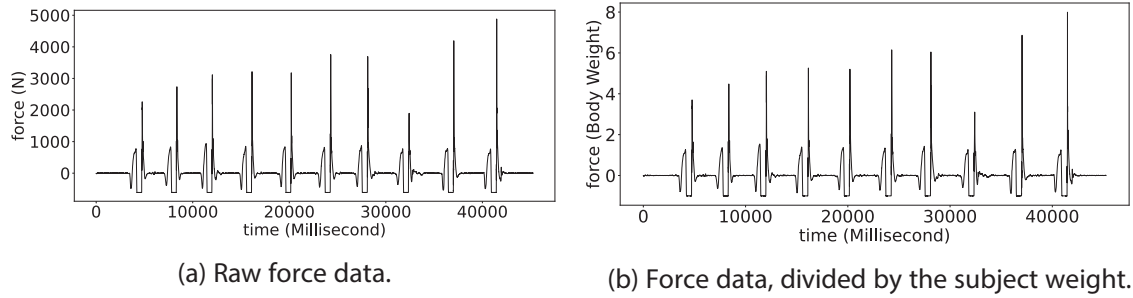


Figure 6: Data preprocessing of CMJ force data. (A) Raw force data. (B) Force data, divided by the subject weight. CMJ, countermovement jump.

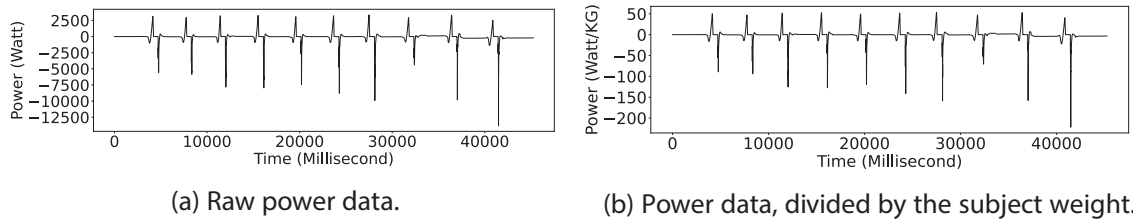


Figure 7: Data preprocessing of CMJ power data. (A) Raw power data. (B) Power data, divided by the subject weight. CMJ, countermovement jump.

calculated, they are ranked in decreasing order. For example, according to Newton's laws of motion, the absolute value of *slope_AC* is positively correlated with peak force, and consequently, it is also positively correlated with flight time. In our experiment, the *F*-scores for *slope_AC* with respect to both peak

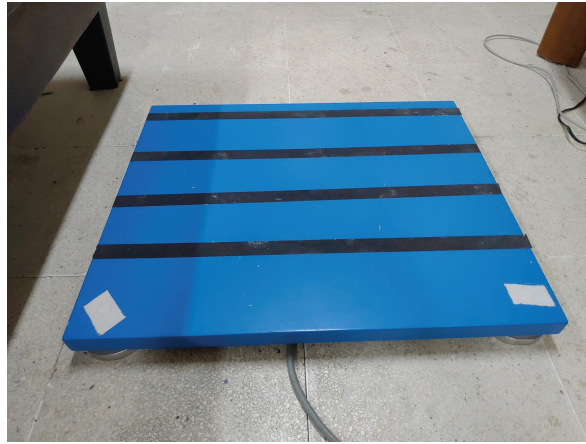
force and flight time are high. Following the ranking results of the 31 features, the top-*k* features with the highest *F*-scores for training are selected. The higher the score, the stronger the relationship between a feature and a performance variable; thus, the *k* most significant features are identified.



(a) Appearance of the SIPPLink Technology NAXSEN IMU.



(b) Placement of the IMU on the test subject.



(c) Kistler 9260AA6 force-plate used during this study.

Figure 8: A NAXSEN IMU tied on a jumper and a Kistler 9260AA6 force plate. (A) Appearance of the SIPPLink Technology. (B) Placement of the IMU on the test subject. (C) Kistler 9260AA6 force plate used during this study. IMU, inertial measurement unit.

d.ii Classical regression models in machine learning

The regression models used in this study include: LSVR, DTR, LR, and several popular ensemble methods, including XGBR and LGBMR. A support vector machine (SVM) is a popular classification method used for machine learning. Unlike deep learning, SVM does not require a large amount of data and offers good performance in many binary classification tasks. LSVR extends SVM when performing regressions. A decision tree (DT) is considered a conventional and widely used machine learning model, which constructs a tree-like data structure, with decision rules, and then infers a result from those rules.

The advantage of DT is its simplicity and explicability, and DTR is an extension of DT used for regressions. LR is a basic regression model for examining issues in both machine learning and statistics. Both light gradient-boosting machine (LGBM) and XGB are well-known ensemble methods, and ensemble methods offer superior predictive ability for many tasks [30]. eXtreme gradient boosting (XGB) is a very powerful ensemble method, which is scalable, and therefore it can fit a range of scenarios [9]. It has also been used in many Kaggle competitions. LGBM is a Microsoft ensemble method that benefits from a shorter training time and higher accuracy, compared to XGB [16]. LGBM can speed up the training process of conventional gradient boosting decision

tree (GBDT) by more than 20 times, while achieving almost the same level of accuracy as XGB.

IV. Model Performance Evaluation

a. Experimental setup

This study simultaneously collects acceleration-time and force-time data. The acceleration-time data are collected using a NAXSEN IMU sensor, produced by SIPPLink Technology.⁴ The size (50 mm × 46 mm × 9 mm) and appearance of the device are shown in Figure 8A. The small dimensions of the device mean that it is easily worn by the test subject, close to their sacrum, as shown in Figure 8B. The force-time data are collected by a 3D portable force plate produced by Kistler (model 9260AA6), and the appearance and dimensions (600 mm × 500 mm × 50 mm) of the device are shown in Figure 8C. The sample rate of both the IMU sensor and force plate is set at 1,000 Hz.

b. Experimental procedure

A total of 19 healthy test subjects aged 20–40 years provided a total of 870 CMJ observations. This placement is closer to the subject's CoM and consequently captures the body dynamics more effectively. Specifically, it provides a lower error rate than when attached to the subject's ankle, according to Lee et al. [22]. The IMU is attached to the test subject's clothing as shown in Figure 8B.

Previous studies [3] report that different types of shoes create a sufficiently large margin of error in the results, which requires compensation, especially affecting the average loading rate and the second peak force when landing. Therefore, each subject performs 10 CMJs barefoot on the force plate, with a short interval between jumps.

The application is configured to capture and log data from the devices at 1,000 Hz, which directly corresponds to the devices' sampling rates. To obtain a data resolution of 1,000 Hz from the IMU, the device must be directly wired to a computer. In this system, a standard universal serial bus (USB) cable is used; however, it should be noted that using a wired connection would be impractical for real-life applications. The only wireless communications protocol offered by the IMU is Bluetooth, which limits the maximum streaming sample rate to 50 Hz. This is inconsistent with the native sampling rate of the force plate,

which is 1,000 Hz. To address this issue, the acceleration-time data are up-sampled from 50 Hz to 1,000 Hz, by both duplication and linear interpolation. The features extracted from the original acceleration-time data (at 1,000 Hz) and the two sets of up-sampled acceleration-time data (now at 1,000 Hz) are used to train the regression models and determine the model fit. Each of the CMJ performance variables is obtained from the force-time data. In total, 80% of the observations are used for model training and 20% are used for testing.

The performance metric used to determine model performance is the mean absolute percentage error (MAPE) model, shown in Eq. (4), where y_k is the actual value of a performance variable, derived from the force-time data; $\hat{y}_k$ is the predicted value of the performance variable, derived by the machine learning method, using the available acceleration-time data; and n is the number of CMJs. Each experiment is repeated five times to ensure the robustness of the model. This study uses the default parameter values for the five regression models in scikit-learn, which are documented in Table 2.

$$MAPE = \frac{100}{n} \sum_{k=1}^n \left| \frac{y_k - \hat{y}_k}{y_k} \right| \quad (4)$$

c. Experimental results

Since using unsuitable features may significantly affect the predictive performance of the selected models, a feature selection task is performed to validate and, where necessary, remedy the model(s). Since there are at most $2^{31} - 1$ non-empty subsets of the 31 extracted features in the worst case, it is time-consuming to exhaustively train different regression models using each of these subsets to obtain the model with the lowest prediction error (e.g. MAPE). Therefore, a feature selection analysis is performed using the scikit-learn Select k Best tool [1] to reduce the exhaustive training effort. For each of the seven performance variables, k is varied from 1 to 31, to select the k most significant features of the related performance variable. The predictive performance for each variable under MAPE is examined using each of the five regression models, with a different k value. The experimental results of each of the seven performance variables are shown in Figures 9A–9G. Figures 9A–9G show that the performance of most regression models improves as the number of k increases, in terms of MAPE. This indicates that

⁴ <https://holdon.siplink.com/>

Table 2: Machine learning parameter settings for the models used in this study

Model	Parameter	Setting value
LSVR [11]	Epsilon	0.0
	Tol	0.0001
	C	1.0
	Loss	epsilon_insensitive
DTR [11]	intercept_scaling	1.0
	max_iter	100,000
	max_depth	None
	min_samples_split	2
	min_samples_leaf	1
	min_weight_fraction_leaf	0.0
XGBR	max_leaf_nodes	None
	min_impurity_decrease	0.0
	learning_rate	0.3
	Subsample	1
	n_estimators	100
	Gamma	0.0
	max_depth	6
	min_child_weight	1
	max_delta_step	0
	num_leaves	31
	max_depth	-1 (no limit)
	Subsample	1
LGBMR	learning_rate	0.1
	n_estimators	100
	min_split_gain	0.0
	min_child_weight	0.001
	min_child_samples	20

DTR, decision tree regression; LGBMR, light gradient boosting machine regression; LSVR, linear support vector regression; XGBR, eXtreme gradient boosting regression.

most of the 31 extracted features are relevant to the predictive ability of the variables.

The experimental results in this section show the comparative success of the five regressions, assessed by MAPE, when estimating the seven CMJ performance variables. These data are tested on the three types of acceleration data: (1) the original 1,000 Hz data, (2) the 50 Hz data up-sampled by duplication, and (3) the 50 Hz data up-sampled by interpolation. These results are shown in Tables 3A–3C, respectively. We can see that LGBMR is the best among all compared regression models to predict almost all of seven performance

variables on all three types of datasets in terms of MAPE. Even though LGBMR is not the best in the task of predicting the average loading rate on the original 1,000 Hz among all models, the MAPE of LGBMR is only slightly inferior to XGBR. Therefore, we still select LGBMR as our final prediction model for its excellent ability to predict almost all of the performance variables of a CMJ. Interestingly, the MAPE in the second peak force is higher than the MAPE in other variables. After consulting with domain experts, we believe that the errors in the second peak force originate from the contact phase, as different subjects may employ varying landing

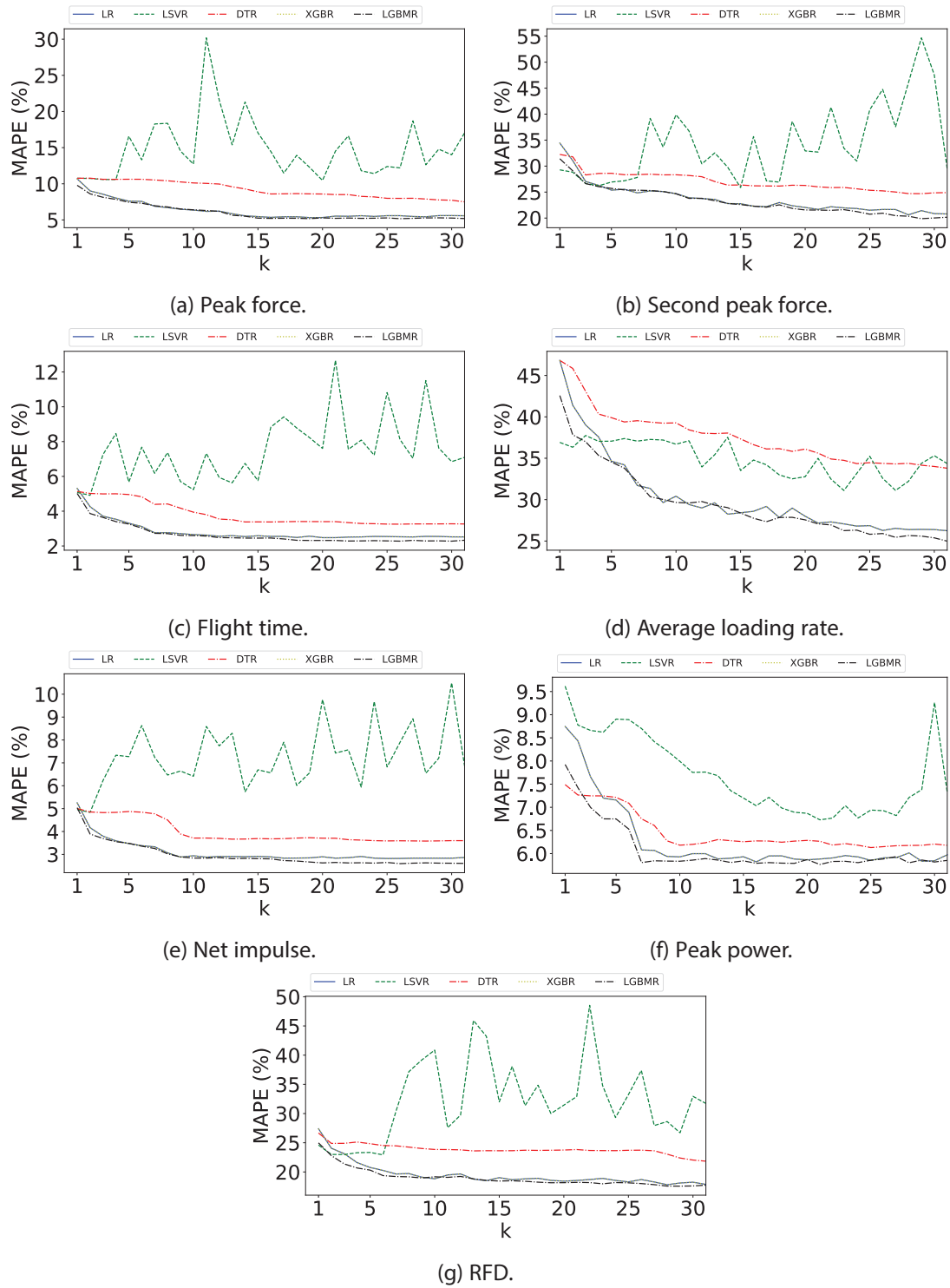


Figure 9: Prediction results from varying the number (k) of selected features for each performance variable, under each of the five regression models. (A) Peak force. (B) Second peak force. (C) Flight time. (D) Average loading rate. (E) Net impulse. (F) Peak power. (G) RFD. DTR, decision tree regression; LGBMR, light gradient boosting machine regression; LR, linear regression; LSVR, linear support vector regression; MAPE, mean absolute percentage error; RFD, rate of force development; XGBR, eXtreme gradient boosting regression.

Table 3: Comparison of MAPE for original data and up-sampling data.

(a) MAPE of different models utilizing the original data (1,000 Hz native).

Variable \ Model	LR [11]	LSVR [11]	DTR [11]	XGBR	LGBMR
Peak force	7.97	10.84	7.54	5.50	5.00
Second peak force	24.88	28.31	23.84	19.16	18.07
Flight time	2.59	4.63	2.85	2.09	2.04
Average loading rate	34.32	33.54	30.68	23.50	23.59
Net impulse	2.99	4.63	3.28	2.61	2.42
Peak power	5.73	6.86	7.21	5.60	5.43
RFD	21.12	23.16	21.73	17.50	16.56

(b) MAPE of different models utilizing up-sampled data (50–1,000 Hz, by duplication).

Variable \ Model	LR [11]	LSVR [11]	DTR [11]	XGBR	LGBMR
Peak force	8.06	11.18	7.60	5.63	5.29
Second peak force	25.84	26.52	28.20	20.96	20.04
Flight time	3.52	4.87	3.22	2.52	2.38
Average loading rate	34.82	33.08	36.44	26.89	26.37
Net impulse	3.82	4.83	3.82	2.89	2.76
Peak power	6.36	7.28	7.84	6.11	5.91
RFD	22.02	23.22	23.65	17.85	17.55

(c) MAPE of different models utilizing up-sampled data (50–1,000 Hz, by interpolation)

Variable \ Model	LR [11]	LSVR [11]	DTR [11]	XGBR	LGBMR
Peak force	7.50	10.27	7.32	5.59	5.18
Second peak force	24.69	26.43	29.32	21.04	19.89
Flight time	3.25	4.80	3.31	2.40	2.28
Average loading rate	33.80	31.72	34.35	26.38	24.99
Net impulse	3.58	4.47	3.71	2.94	2.60
Peak power	6.13	6.67	7.25	5.83	5.76
RFD	21.84	22.94	23.70	18.00	17.57

The bold values indicate the smallest MAPE among the five models.

DTR, decision tree regression; LGBMR, light gradient boosting machine regression; LR, linear regression; LSVR, linear support vector regression; MAPE, mean absolute percentage error; RFD, rate of force development; XGBR, eXtreme gradient boosting regression.

strategies. For each performance variable of a CMJ, the selected features achieving the best prediction results of LGBMR are listed in Table 4.

d. Data validation

So far, in this study, LGBMR has proven to be the superior machine learning-based method for

Table 4: Data features used to predict values for the seven CMJ performance variables

Feature	Peak force	Second peak force	Flight time	Average loading rate	Net impulse	Peak power	RFD
Mean	V	V	V	V	V	V	
SD	V	V	V	V		V	V
IQR	V	V	V	V	V	V	V
Skewness	V	V	V	V	V	V	V
Kurtosis		V	V	V			V
Frequency	V	V	V	V			V
Entropy	V	V	V	V	V		V
value_A	V	V	V	V	V	V	V
value_C	V		V	V	V	V	V
value_E	V	V	V	V	V	V	V
value_F	V	V	V	V	V		V
duration_AC		V	V	V	V	V	V
duration_BD	V	V	V	V	V	V	V
duration_CE	V	V	V	V	V	V	V
duration_EF		V	V	V	V	V	V
slope_AL	V	V	V	V	V	V	V
slope_AR	V	V	V	V	V	V	V
slope_CL	V	V	V	V	V	V	V
slope_CR	V		V	V			V
slope_EL		V	V	V	V	V	
slope_ER	V	V	V	V	V	V	V
slope_FL	V	V	V	V	V	V	V
slope_FR	V	V		V			V
slope_AC	V	V	V	V	V		V
slope_CE	V	V	V	V	V		V
slope_DE		V	V	V	V		V
slope_EF	V	V	V	V	V		V
Integration	V	V	V	V	V	V	V
average_A	V	V	V	V	V	V	V
average_C	V	V	V	V	V	V	V
average_E	V	V	V	V	V	V	V

CMJ, countermovement jump; RFD, rate of force development; SD, standard deviation.

predicting flight time. However, there are alternate conventional ways of collecting acceleration data to estimate the vertical jump variable. To compare the predictive performance of the LGBMR model being examined here, it is now compared to the performance of the integration-based method proposed in reference [27]. As shown in Table 5, in terms of MAPE, LGBMR outperforms the integration-based method, when predicting CMJ flight time.

V. Implementation of Demonstration System

A test subject demonstration is used to test the predictive ability of the system, using a commercially available IMU sensor, connected to a computer via Bluetooth. The IMU sensor is manufactured by Rabboni⁵ and produced by SIPPLink Technology. The dimensions (44 mm × 44 mm × 15 mm) and

appearance are shown in Figure 10A. As described previously, the device is affixed to the test subject’s sacrum area, as shown in Figure 10B. The sample rate of the IMU is 50 Hz over Bluetooth, and the data needs to be up-sampled to 1,000 Hz by interpolation, following the experimental result discovered in Section IV.

The system architecture and application workflow of the demonstration system are exhibited in Figure 11. First, a file name is created so that the acceleration data are correctly saved. Then, the computer connects to the IMU sensor via Bluetooth. If the connection is successful, the application begins to show real-time acceleration-time data from the IMU (Figure 12, bottom), and then the test subject can perform a jump. The system will determine whether the type of jump is a CMJ; if it is, the system applies LGBMR to predict each of the seven performance variables associated with a CMJ. The predicted value of each performance variable is then displayed

Table 5: Relative predictive performance of the CMJ flight time variable, measured using MAPE

	Integration-based method	LGBMR
CMJ	11.75	2.04

The bold values indicate the smallest MAPE among the five models.
CMJ, countermovement jump; LGBMR, light gradient boosting machine regression; MAPE, mean absolute percentage error.



(a) Commercially available Rabboni IMU device. (b) Placement of the device on a test subject.

Figure 10: Rabboni IMU tied on a jumper. (A) Commercially available Rabboni IMU device. (B) Placement of the device on a test subject. IMU, inertial measurement unit.

⁵ <https://rabboni.com.tw/about-rabboni/>

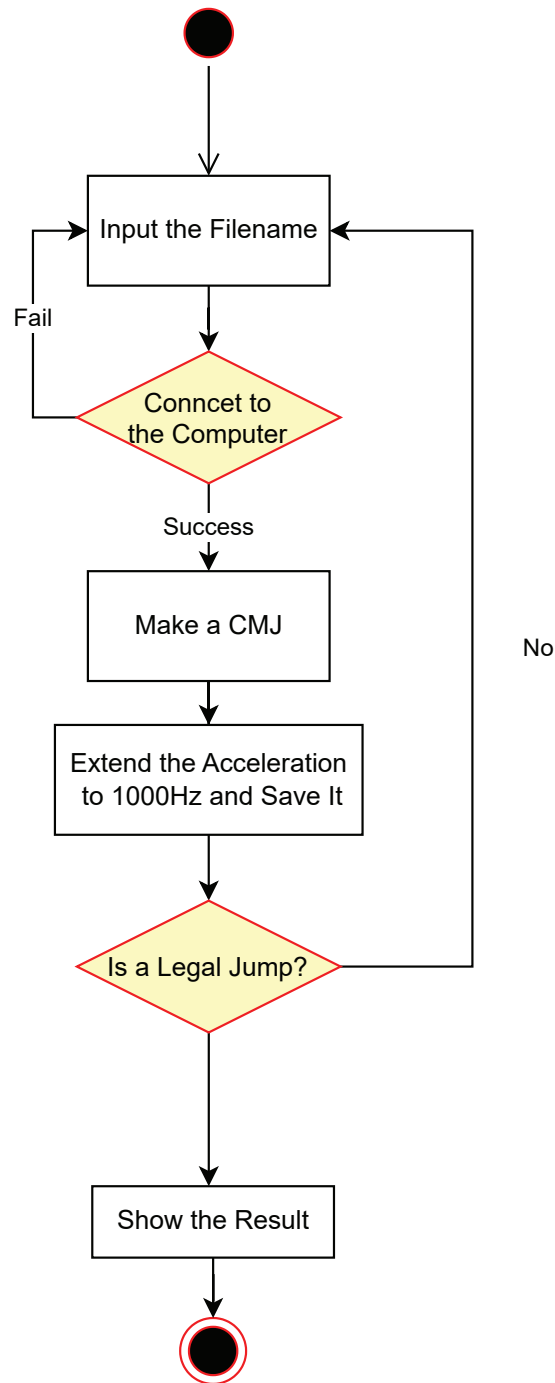


Figure 11: System architecture and application workflow. CMJ, countermovement jump.

(Figure 12, top). Note that in all tests the IMU is positioned close to the sacrum, as stated in references [3, 22], and the test subjects are barefoot. If the IMU is installed in a different location, or if the test subjects are wearing shoes, the algorithm may need to be redesigned to reduce the error range.

VI. Conclusions

This paper demonstrates a method of estimating each of the seven performance variables used to assess a CMJ. It is shown that this is possible using an economical, wearable IMU device, combined with a machine

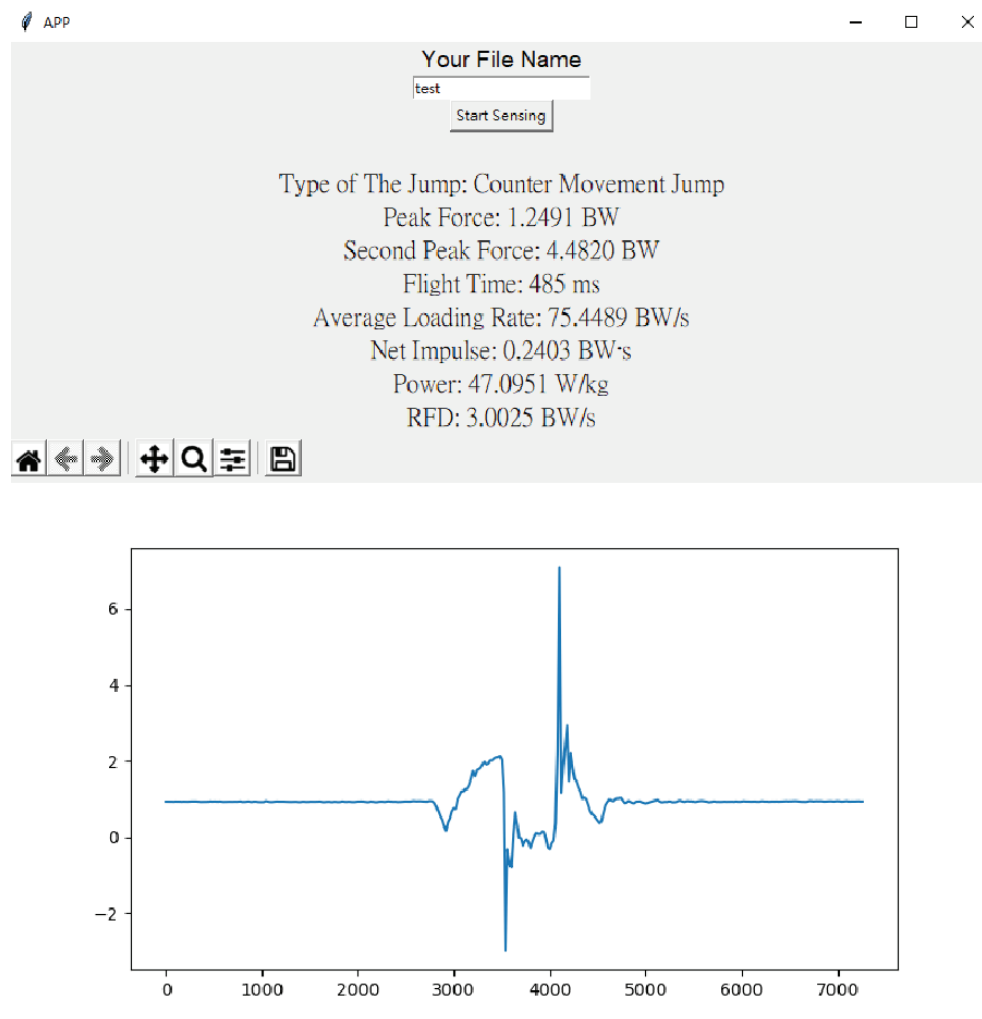


Figure 12. Prototype system predicting CMJ variable performance in a live test. CMJ, countermovement jump.

learning system. For the everyday consumer, this is a viable and (in many cases) preferable alternative to the expensive and cumbersome experience that involves laboratory equipment and force plates. However, the system demonstrated here has certain limitations.

This experiment showed that most of the seven performance variables are reliably captured using an LGBMR-based IMU sensor, including peak force, flight time, net impulse, and peak power. Moreover, this study develops a system for simplifying the measurement process, which is easy to use for persons involved in sports-related sciences. Most importantly, it is clear that the performance variables of a CMJ can be accurately predicted by lightweight, wearable devices when using the correct machine learning model, in this case, LGBMR.

a. Limitations and further research

Despite the progress made in this study, there is considerable scope for further research. For example, rather than up-sampling the acceleration-time observations, it is likely possible to derive more findings from the available data; however, this will take time as it involves hunting for patterns and features.

Furthermore, obtaining a larger sample of test subjects would be beneficial, as this would provide a more comprehensive evaluation of the system across a broader range of body weights, fitness levels, and demographic groups. In particular, it will be necessary to collaborate with athletes and coaches to assess key aspects, such as latency, accuracy, usability, and overall applicability in real-world settings. Such collaboration is also essential to deepen our

understanding of the precise relationships among the variables investigated in this study.

Acknowledgment

This work was supported in part by the National Science and Technology Council, Taiwan, under contracts 111-2222-E-224-003-MY2, 112-2221-E-A49-115, 113-2425-H-A49-001, 113-2221-E-A49-165-MY3 and 113-2221-E-224-044.

References

- [1] sklearn.feature_selection.selectkbest. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html. Last Accessed: 2021-10-9.
- [2] What is an inertial measurement unit? Available at: <https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu>. Last Accessed: 2021-10-9.
- [3] Arnel Aguinaldo and Andrew Mahar. Impact loading in running shoes with cushioning column systems. *Journal of Applied Biomechanics*, 19(4):353–360, 2003.
- [4] Gobinath Aroganam, Nadarajah Manivannan, and David Harrison. Review on wearable technology sensors used in consumer sport applications. *Sensors*, 19(9):1983, 2019.
- [5] VV Bauman and SCE Brandon. Preliminary investigation of predicting time-to-next heel-strike using accelerometers and machine learning. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 827–832. IEEE, 2020.
- [6] Jamin Casselman, Nicholas Onopa, and Lara Khansa. Wearable healthcare: Lessons from the past and a peek into the future. *Telematics and Informatics*, 34(7):1011–1023, 2017.
- [7] Courtney R Chaaban, Nathaniel T Berry, Courtney Armitano-Lago, Adam W Kiefer, Michael J Mazzoleni, and Darin A Padua. Combining inertial sensors and machine learning to predict vgrf and knee biomechanics during a double limb jump landing task. *Sensors*, 21(13):4383, 2021.
- [8] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.
- [9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.
- [10] Yi-Yu Chiang. Replacing force plate with wearable imu for measuring countermovement jump performance with machine learning methods. *Master Thesis of National Chiao Tung University*, 2020.
- [11] Yi-Yu Chiang, Wen-Yueh Shih, Wei-Han Chen, Jiun-Long Huang, and Tzzy-Yuang Shiang. A machine learning-based countermovement performance measurement method using a wearable imu. In *Proceedings of International Conference on Pervasive Artificial Intelligence (ICPAI)*, 2020.
- [12] Luis A Durán-Vega, Pedro C Santana-Mancilla, Raymundo Buenrostro-Mariscal, Juan Contreras-Castillo, Luis E Anido-Rifón, Miguel A García-Ruiz, Osval A Montesinos-López, and Fermín Estrada-González. An iot system for remote health monitoring in elderly adults through a wearable device and mobile application. *Geriatrics*, 4(2):34, 2019.
- [13] Raffaele Iervolino, Francesco Bonavolontà, and Adolfo Cavallari. A wearable device for sport performance analysis and monitoring. In *2017 IEEE International Workshop on Measurement and Networking (M&N)*, pages 1–6. IEEE, 2017.
- [14] Daniel A James and Nicola Petrone. *Sensors and wearable technologies in Sport: Technologies, trends and approaches for implementation*. Springer, 2016.
- [15] William R Johnson, Ajmal Mian, Mark A Robinson, Jasper Verheul, David G Lloyd, and Jacqueline A Alderson. Multidimensional ground reaction forces and moments from wearable sensor accelerations via deep learning. *IEEE Transactions on Biomedical Engineering*, 68(1):289–297, 2020.
- [16] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30:3146–3154, 2017.
- [17] Suwon Kim and Seongcheol Kim. A multi-criteria approach toward discovering killer iot application in korea. *Technological Forecasting and Social Change*, 102:143–155, 2016.
- [18] Kristof Kipp, John Krzyszkowski, and Daniel Kant-Hull. Use of machine learning to model volume load effects on changes in jump performance. *International Journal of Sports Physiology and Performance*, 15(2):285–287, 2020.
- [19] Tyler J Kirby, Jeffrey M McBride, Tracie L Haines, and Andrea M Dayne. Relative net vertical impulse determines jumping performance. *Journal of Applied Biomechanics*, 27(3):207–214, 2011.
- [20] Jason Lake, Peter Mundy, Paul Comfort, John J McMahon, Timothy J Suchomel, and Patrick Carden.

Concurrent validity of a portable force plate using vertical jump force–time characteristics. *Journal of Applied Biomechanics*, 34(5):410–413, 2018.

[21] Jason P Lake, Peter D Mundy, and Paul Comfort. Power and impulse applied during push press exercise. *The Journal of Strength & Conditioning Research*, 28(9):2552–2559, 2014.

[22] Huang-Chen Lee, Soun-Cheng Wang, and Zih-Hua Lin. An open-source wearable sensor system for measuring the duty factor of runners. *IEEE Transactions on Instrumentation and Measurement*, 73, 2024.

[23] He Li, Jing Wu, Yiwen Gao, and Yao Shi. Examining individuals’ adoption of healthcare wearable devices: An empirical study from privacy calculus perspective. *International Journal of Medical Informatics*, 88:8–17, 2016.

[24] Hyerim Lim, Bumjoon Kim, and Sukyung Park. Prediction of lower limb kinetics and kinematics during walking by a single imu on the lower back using machine learning. *Sensors*, 20(1):130, 2020.

[25] Christopher P McLellan, Dale I Lovell, and Gregory C Gass. The role of rate of force development on vertical jump performance. *The Journal of Strength & Conditioning Research*, 25(2):379–385, 2011.

[26] John J McMahon, Shannon Murphy, Sophie JE Rej, and Paul Comfort. Countermovement- jump-phase characteristics of senior and academy rugby league players. *International Journal of Sports Physiology and Performance*, 12(6):803–811, 2017.

[27] Pietro Picerno, Valentina Camomilla, and Laura Capranica. Countermovement jump performance assessment using a wearable 3d inertial measurement unit. *Journal of Sports Sciences*, 29(2):139–146, 2011.

[28] Juliano Dal Pupo, Daniele Detanico, and Saray Giovana dos Santos. Kinetic parameters as determinants of vertical jump performance. *Revista Brasileira de Cineantropometria & Desempenho Humano*, 14:41–51, 2012.

[29] Paige E Rice, Courtney L Goodman, Christopher R Capps, N Travis Triplett, Travis M Erickson, and Jeffrey M McBride. Force-and power-time curve comparison during jumping between strength-matched male and female basketball players. *European Journal of Sport Science*, 17(3):286–293, 2017.

[30] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[31] Paul Swinton, Justin Keogh, and Jason Lake. Practical applications of biomechanical principles in resistance training: The use of bands and chains. *Journal of Fitness Research*, 3:26–41, 2014.

[32] Kuniharu Takei, Wataru Honda, Shingo Harada, Takayuki Arie, and Seiji Akita. Toward flexible and wearable human-interactive health-monitoring devices. *Advanced Healthcare Materials*, 4(4):487–500, 2015.

[33] Dong Wen, Xingting Zhang, and Jianbo Lei. Consumers’ perceived attitudes to wearable devices in health monitoring in china: A survey study. *Computer Methods and Programs in Biomedicine*, 140:131–137, 2017.

[34] Zhaoxian Zhou, Sarbagya Shakya, and Zhanxin Sha. Predicting countermovement jump heights by time domain, frequency domain, and machine learning algorithms. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 167–170. IEEE, 2017.